

N87-16778 <sup>D</sup>36-63  
20P  
69652

1986

**NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM**

**MARSHALL SPACE FLIGHT CENTER**

**APPLICATIONS OF ARTIFICIAL INTELLIGENCE TO  
SCIENTIFIC RESEARCH**

Prepared by:	Mary Ellen Prince, M.S.
Academic Rank:	Instructor
University and Department:	University of Alabama in Huntsville Department of Computer Science
NASA/MSFC:	
Laboratory:	Space Sciences
Division:	Astrophysics
Branch:	X-ray Astronomy
MSFC Colleague:	Martin C. Weisskopf
Date:	August 8, 1986
Contract No.:	NGT 01-002-099 The University of Alabama

# APPLICATIONS OF ARTIFICIAL INTELLIGENCE TO SCIENTIFIC RESEARCH

by

Mary Ellen Prince  
Instructor in Computer Science  
University of Alabama in Huntsville  
Huntsville, Alabama

## ABSTRACT

Artificial intelligence (AI) is a growing field which is just beginning to make an impact on disciplines other than computer science. While a number of military and commercial applications have been undertaken in recent years, few attempts have been made to apply AI techniques to basic scientific research. This study will show that there is no inherent reason for the discrepancy. The characteristics of the problem, rather than its domain, determines whether or not it is suitable for an AI approach. Expert systems, intelligent tutoring systems, and learning programs are examples of theoretical topics which can be applied to certain areas of scientific research. Further research and experimentation should eventually make it possible for computers to act as intelligent assistants to scientists.

## INTRODUCTION

Artificial intelligence (AI) is an area of computer science which has recently begun to generate a great deal of interest as government and industry alike embark on ambitious programs designed to apply AI techniques to real world situations. At a time when professionals in other areas are being encouraged to adopt this new approach to computer problem solving, debate continues regarding its possible usefulness. Some proponents are inclined to make extravagant claims about the prospects for applied AI. While this enthusiasm is commendable, there is nevertheless a danger that it might produce unrealistic expectations on the part of prospective AI users. There is, on the other hand, a persistent group of skeptics who believe there is little possibility that scientists will be able to build systems capable of intelligent behavior, at least using current technology. Yet a third group is composed of people who are unfamiliar with the current state of AI and are thus unaware of the potential benefits that it have for them. To help resolve these conflicting opinions it is important that AI scientists begin to provide realistic appraisals of the scope and the limitations of artificial intelligence.

The purpose of this study is to lay the groundwork for just such an appraisal. In particular, it concentrates on the possibilities of applying AI to other areas of scientific endeavor. Section I will attempt to indicate the kinds of problems that are particularly suitable to AI techniques. Section II discusses some basic subfields of AI that are applicable to scientific research. The conclusion of the study will indicate directions for future work.

## OBJECTIVES

The objectives of this study were

1. To provide an overview of the field of Artificial Intelligence and to differentiate the capabilities provided by this branch of Computer Science from those provided by more traditional computational methods.
2. To identify techniques and research areas within the AI field which are applicable to scientific research.
3. To identify specific applications where these approaches may prove useful.

## SECTION I

Defining artificial intelligence is, in some respects, as difficult as defining "real" (or human) intelligence. Theoreticians may beg the question by saying that artificial intelligence is the use of computers to model human intelligence, or that it is the study of the relationship between cognition and computation. Potential users of AI technology need a more practical statement than this. Let us begin, then, by saying that artificial intelligence is a branch of computer science that solves problems which cannot be easily solved using traditional computational methods. In other words, AI problems are somehow different from traditional problems and must therefore be solved differently. What are some of the distinguishing characteristics of an AI problem?

### AI problems are non-algorithmic

Strong mathematical models exist for most traditional problem areas. Consequently, algorithms can be found which describe solution procedures exactly. AI is most useful in loosely structured domains where no clear cut procedures exist. Problem solving in these domains can be characterized as a search for a goal state, where in this case a goal state is a description of some desired world situation [Hayes-Roth, 84]. The solution a program produces, then, is the path it takes to achieve the desired goal. A chess playing program, for example, has its goal a victory, and the path to that goal is the series of moves it makes in playing a particular game.

In the absence of well defined procedures for achieving goals the AI programmer may employ heuristic problem solving methods. A heuristic is a rule of thumb, an embodiment of common sense or intuitive knowledge. Heuristics do not, in general, guarantee optimal solutions to problems. What they do provide is a plausible approach for finding a reasonable solution in a reasonable amount of time. Heuristic-based problem solving can also be useful in domains where there are existing algorithms which describe techniques for exhaustively enumerating all possible solution paths. Finding the desired solution requires testing each path until one is found which leads to a goal state. The difficulty is that exhaustive searches of this nature are often computationally infeasible. Here, heuristics can be used to constrain the set of possible solutions to a subset consisting of only the most feasible possibilities.

## AI problems are symbolic instead of numeric

One of the most important characteristics of an AI program is its ability to manipulate arbitrary symbols. While traditional scientific computation is dominated by mathematical operations, artificial intelligence deals with qualitative measures and objects whose values may be non-numeric. For example, a medical diagnostic program must be able to represent concepts such as "low grade fever" or "primary cause of infection".

## AI problems require large amounts of knowledge

"Knowledge" is not synonymous with "facts". A computer can store tremendous amounts of factual information, but the value of this information depends on the ability of computer programs to use it well. To this end, AI researchers continue to look for new and better methods of knowledge representation. Frames, scripts, semantic networks, and rule structures are all attempts to organize factual data into true knowledge so AI programs will be able to make plausible inferences, derive analogies, and, in general, exhibit the kind of behavior which in humans we would consider intelligent. Because of the broad scope of many AI problem domains and the lack of good domain models, much of this knowledge is embodied in heuristics.

It should be noted at this point that artificial intelligence is an elusive concept. Some AI programs are based on strong mathematical models, some employ algorithmic methods, and some do not require significant knowledge bases. On the other hand, there are programs which exhibit all of these characteristics and yet cannot be said to be intelligent.

## SECTION II

Basic research in artificial intelligence is concentrated in a number of areas. Expert systems, computer vision, robotics, and knowledge representation are just a few. A complete survey is beyond the scope of this paper. Instead, several topics with potentially useful applications in scientific research have been chosen for discussion. A description of each topic will be given, along with a summary of some interesting applications.

### Expert Systems

The current high level of interest in artificial intelligence can be traced in large part to the commercial

success of expert systems. For almost the first time AI has emerged from the research laboratory with a product that is capable of solving real world problems. Expert systems perform tasks that would normally require the knowledge, experience, and intuition of an expert. They can be distinguished from ordinary programs which embody expertise in several ways. One of the main differences is that the program is structured in a non-traditional manner.

An expert system consists of a working memory, a knowledge base, and a procedural portion (frequently referred to as an "inference engine"). The working memory is similar to the data section of a traditional program. It contains facts specific to a particular instance of the problem. The knowledge base is a collection of domain facts and heuristics, frequently expressed as "rules", or situation-action pairs. The inference engine determines when and how the rules are to be executed, or "fired". The virtues of this organization are two-fold. Separation of domain knowledge from methods of applying that knowledge allows the same inference engine to be used for multiple tasks. This is the basis of the expert system "shell" concept. Furthermore, since the knowledge base is a loosely structured collection of rules, with control information restricted to the inference engine, it can be developed incrementally and thus can grow as knowledge in a given field grows and is distilled into expertise.

What types of problems are candidates for expert system solution? In general, where simple algorithmic approaches exist, they should be used. The exception to this lies in cases where algorithms generate so many potential solutions that exhaustive search is unacceptably expensive. Additionally, there must be a domain expert who is able to contribute his know-how to the project. The knowledge an expert brings to an expert system can be broken down into three levels: declarative knowledge, readymade or experiential knowledge, and meta-knowledge [Hong, 86].

Declarative knowledge consists of domain concepts and their interrelationships. In theory, this type of knowledge should enable the expert system to perform "deep reasoning", or reasoning from first principles. In practice, few expert systems are capable of such deep reasoning, although this is a problem of continuing research interest. The difficulty lies in the lack of adequate system models and the expense of deriving solutions in this manner.

Experts do not usually employ deep reasoning to solve problems. Rather they rely on ready-made, or empirical, knowledge to rapidly arrive at solutions, drawing analogies from past experiences and recalling shortcuts. Reasoning

from first principles is reserved for new situations. Most state-of-the-art expert systems employ this kind of "shallow" reasoning.

Meta-knowledge, or knowledge about knowledge, is the essence of expertise. In expert systems, this amounts to knowing when and how to apply the specific rules. The ability to recognize in a current situation analogies to past situations, to know when certain shortcuts can be fruitfully applied, is vital to effective problem solving.

Current applications of expert system technology span a broad spectrum. Some of the best known are discussed briefly.

Mycin, a medical diagnostic system, is able to diagnose and treat infectious blood diseases [Buchanan and Shortliffe, 84].

R1 (recently renamed XCON) is used by Digital Equipment to configure VAX computer components [McDermott, 81].

DENDRAL determines the molecular structure of an unknown molecule based on its mass spectrographic analysis [Barr and Feigenbaum, 82].

Because it is an example of how expert system technology can be used to assist research scientists, we will examine DENDRAL in more detail. The problem statement is as follows: Given the spectroscopic analysis of an unknown molecule with known constituent atoms, determine its molecule with known constituent atoms, determine its molecular structure. A known algorithm (DENDRAL) exists which will enumerate all possible acyclic structures, given the constituent atoms. Heuristic DENDRAL uses additional data from the mass spectrographic analysis to derive a set of constraints, which are inferred from heuristic rules provided by expert chemists. Following this, the DENDRAL algorithm is used to generate only those structures which satisfy the constraints, greatly reducing the amount of computation required. Finally, the structures thus generated are tested by being run through a simulated mass spectrometer. By comparing the simulated spectra to the actual data, the most likely structure can be determined.

The DENDRAL project is an ongoing research effort. Over the years it has evolved into a more general system than the original version. One improvement, Meta-DENDRAL, added a learning element which allowed the program to "learn" new rules describing the operation of the mass spectrometer [Cohen and Feigenbaum, 82].



## Intelligent Tutoring Systems

Computer-aided instruction (CAI) is not new. In its earliest form it provided little more than an electronic textbook accompanied by a set of problems which were used to drill the student. Intelligent tutoring systems (ITS) are designed to surpass this model and furnish the same kind of individualized instruction that would be expected from an experienced human tutor.

A complete ITS has three components: the expert, the tutor or coach, and the student modeler. Because of the complexity of the issues involved, most current research focuses on one or two of these elements.

The expert contains the knowledge base of the system. It should be capable of generating problems that are tailored to the strengths and weaknesses of the individual, and, ideally, should be able to solve problems that the student poses. Whenever possible, it should have the ability to offer explanations for its actions.

The student modeler analyzes responses to discover weaknesses in the student's understanding. Overlay models express his knowledge as a subset of the expert's knowledge. By comparing the learner's solution to one generated by the expert, gaps in his mastery of the subject matter can be identified and corrected. Other techniques attempt to represent the student as a collection of "bugs", or misconceptions, while still others try to measure his ability by determining his location in a graph or heirarchical network of skills.

The tutor controls communication with the student. It points out errors and misconceptions, poses problems to be solved, and offers factual information where it is needed. In an ideal situation, the tutor will not only recognize that the student has made an error, but will understand why that error was made. It can thus guide the student through the problem solving process in such a way that he will recognize his own mistakes.

Intelligent tutoring systems have been implemented on a variety of domains.

WHY [Stevens, et al., 82] uses the Socratic dialog technique to help students understand the causal relations which produce heavy rainfall in certain climates.

GUIDON [Clancey 79, 82] instructs medical students in the diagnosis and treatment of infectious blood diseases. It uses Mycin, an expert system, as its expert component. Mycin has been augmented by additional information to allow the tutor to answer questions and provide explanation.

SOPHIE [Brown, et al. 82] uses an internal simulator to tutor troubleshooting techniques for electronic devices. It allows students to experiment with the device actively, by modifying the simulation.

LEVI [Matthews and Biswas, 85] acts more like an intelligent assistant than a tutor. It monitors users of a computer screen editor running under the UNIX operating system. Using knowledge it has collected about the user's level of expertise, LEVI makes suggestions which are designed to enable him to utilize the system more effectively.

The ITS concept offers great promise. Technology has increased, rather than decreased, the need for skilled personnel. Computer based training programs can play an important role in providing this personnel. Computers can function as coaches, assistants, or lab instructors. Many problems remain to be solved, however. Psychologists and AI researchers must collaborate to determine effective tutoring strategies. Domain independent theories of student modeling need to be derived. One of the severest limitations on current systems is the lack of good natural language processing techniques. Open communication between student and teacher rely on menu-driven communication or on simple command languages based on natural language. This limitation restricts the student's ability to pose new and unexpected questions and to engage in discovery learning.

### Learning

Learning is one of the chief earmarks of intelligence. Without it, a system is purely mechanistic. When confronted with a set of circumstances it will always respond in the same way, no matter how poorly this response has worked in the past. It will never have a new idea, or modify its view of the world. Traditional computer programs fall into this model.

A major thrust of AI research has been to build into computers the learning abilities that we take for granted in a human being. To date, the results of this effort have been mixed, but promising techniques and innovative approaches hold out hope for the future.

Many different kinds of cognitive activity can be classified as learning. Rote memorization, acquiring and improving skills, and learning from examples are all facets of the broad topic and have been investigated at one time or another by researchers in artificial intelligence. Learning can be directed toward improving performance in a particular task, or it can have as its objective the general acquisition of knowledge and the integration of this knowledge into a coherent domain model. AI research has emphasized learning to improve performance although recently some interesting work has been done in the latter area.

Learning is a very complicated process, not fully understood even by psychologists. Early researchers in artificial intelligence hoped to be able to discover some general purpose learning mechanism. A computer provided with this mechanism could then "learn", instead of being programmed, much like a human baby learns through interaction with its environment. Unfortunately, attempts to model the human brain and its learning processes were largely unsuccessful. One of the most significant results to come out of this early work was the realization that learning does not occur in a vacuum. To learn complex concepts, a system must already possess a large body of related knowledge. Learning involves modifying both the structure and content of this knowledge base. Efficient knowledge representation techniques, therefore, are as important to this area of research as they are to expert systems.

To date, AI researchers have concentrated on observational, or inductive, learning. By repeatedly observing events in its environment the program is able to infer general principles and thus acquire new knowledge. The events may even be attempts by the program itself to perform some task, in which case it is able to modify its own performance.

Learning experiments are usually conducted in tightly controlled environments, for reasons that will be pointed out later. The program is supplied with a set of basic definitions and relations. Then it is repeatedly presented with examples and non-examples of the concept to be learned. Gradually it derives a set of distinguishing features that are necessary and sufficient to define the concept. The program is successful when it is able to use its own concept definitions to classify examples correctly.

For simplicity, most programs of this kind focus on learning one concept at a time. Winston's well-known and influential research on the learning of structural concepts is a good example. His program operated in a world of three

dimensional blocks, where it learned to recognize structures such as arches. The techniques developed during this research have become the basis for much of the later work in learning theory.

Many tasks, of course, require knowledge of a set of concepts. Here the situation becomes more complicated, as the learning element must build definitions that are able to discriminate reliably among the various concepts.

All concepts learning programs must cope with certain inherent difficulties. Chief among these is the choice of the examples (sometimes called the "training set") which will be provided to it. It is important that the training set contain enough positive instances to describe every necessary feature. Negative instances are also vital to delimit the boundaries of the concept. If a program is trying to learn to recognize "red rectangle" it is not enough to present it with red rectangles of varying dimensions. It must have, for example, a red triangle and a blue square as negative instances in order for it to be sure that both color and shape are essential to the concept. In a limited domain with instances chosen and classified by an external teacher, this is not an insurmountable problem. As the learning environment expands, or in the absence of a trainer, difficulties become apparent.

The situation-identification problem is an illustration of these difficulties [Charniak and McDermott, 85]. In extracting the pertinent features of a concept definition the computer must be able to determine what is relevant and what can be safely ignored. In Winston's arch learning program, for example relevant features include number of constituent parts (three), and relative position of the parts (two non-touching supports, one crosspiece). Negative training instances can be used to show that the size and color of the blocks are not essential to the concept. If, however, all of the known examples of arches are embedded in larger scenes, the program must somehow know that it can ignore large portions of the data entirely. If it does not have the ability to discriminate between essential and irrelevant factors, it can become forever bogged down in senseless detail.

Ambiguities in the training set can also be caused by errors in interpretation. For example, if instances are presented visually, they must be transformed into some internal representation that the computer can understand. This transformation process is sometimes quite difficult, and if not performed correctly may produce data that is noisy and unreliable.

Finally, we must consider how to handle situations where there is no trainer to select training instances. In this identifying case, the program must have a body of heuristics to aid it in identifying appropriate examples. It must be provided with feedback to allow it to check its results. It may be forced to settle for less than complete certainty in its results.

Rule learning programs are in some respects quite similar to concept learning programs. The program is initialized with a set of rules which may be used as operators in a particular domain. As in expert systems, rules are usually expressed as situation-action pairs, or alternatively as hypothesis-conclusion pairs. In the most general case this rule set may be incomplete and it may contain incorrect rules. The job of the program is to learn how to perform certain tasks in the given domain. This is accomplished by applying the rules according to some pattern and observing the results. In situations where the rule set is complete, the program must learn which rules to use for a given task and the order in which they should be applied. The general case may also require that rules be added to the set or that existing rules have their hypotheses modified to guarantee their correct application.

Rule learning programs have two parts: the critic and the modifier [Bundy et al., 85]. The critic has the responsibility of determining when a rule has fired incorrectly. In complex tasks, where the interaction between rules is not fully understood, this can be quite difficult. Knowing that a program has produced erroneous results is not the same as being able to identify the source of the error. AI scientists call this the credit assignment problem, and it has been the subject of a great deal of research. Once the error has been isolated, it is still not always clear what changes must be made to produce correct results. Possible corrections include modifying the order in which the rules fire, adding additional conditions to a rule's hypothesis to limit its applicability, and adding new rules to the set.

To date, computer learning techniques have not found widespread real world application. Some of the reasons for this are: problem domains, of necessity, must be relatively narrow and well defined; knowledge representation techniques lack the flexibility some tasks require; methods of "remembering" and applying previously acquired knowledge are still limited. Nevertheless, research has progressed to the point where promising results can be expected in the near future.

Of all the single-concept learning programs Winston's Block World is probably the best known. Another interesting example is Langley's BACON, a set of programs which learned rules relating real-valued variables [Langley, 79]. When given empirical data and some information regarding the dependency relations between variables, BACON was able to learn (or "discover") a number of fundamental scientific principals; e.g. Ohm's law and Kepler's law. BACON was limited in its ability to deal with symbolic concepts, however, and was extremely sensitive to noisy data and to the order in which the training instances were presented.

Multiple-concept learning is exemplified by Meta-DENDRAL. This is also an example of a learning program which has been successful in application, as well as in learning theory research. Recall that heuristic DENDRAL relies on a mass-spectrometer simulator to test proposed molecular structures. The simulator uses a set of cleavage rules to predict which chemical bonds in the structure will be broken, thus producing a simulated mass spectrum. Different structural families of molecules exhibit different cleavage rules.

Meta-DENDRAL is given a set of known molecules from a single structural family, their structures, and their mass spectra. From this data it can infer the cleavage rules for this specific structural family. Using heuristics supplied by chemists and some theoretical knowledge of how mass spectrometers work, the program generates a set of hypotheses which are tested against the training set. Repeated applications of this process produce an approximate set of rules which are then further refined.

Research in rule-learning is still in the early stages. Most of the programs in this category are relatively limited in scope. LEX learns to perform symbolic integration [Mitchell, 77]. Its rules consist of a set of integration and simplification operators; its goal is to develop heuristics that will guide it in the application of the rules to actual problems. One of the distinguishing features of LEX is its ability to propose experiments (in the form of problems to be solved) that will help it refine its procedures.

Most of the inductive learning projects described suffer from certain inherent limitations. They are task-oriented, in that they have been told what they are supposed to learn. They depend upon external sources to provide the data which guides the learning process. They are not well equipped to generate new ideas.

Imagine instead a program which is able to exercise creative control over its own operation, a program whose purpose is to explore a new domain of knowledge guided only by its own evaluation of what is interesting. A learning program of this kind should be able to hypothesize new theories and propose experiments to test the theories. The AM/EURISKO project [Lenat, 83a, 83b] is an exciting and innovative attempt to construct a computer program according to this model.

AM, the first program in Lenat's project, explored the use of heuristics to guide empirical theory formation in a variety of domains. EURISKO extended the work done by AM to include the generation of new heuristics. Both programs were designed to investigate inductive reasoning in the process of scientific research. Lenat based his work on what he calls the accretion model of theory formation. Briefly stated, the model is as follows:

1. For each domain which will be considered, provide the program with an initial set of definitions, operations, and rules.
2. Gather empirical data: examples of rules and definitions, applications of operators, etc.
3. Look for patterns and exceptions in the data.
4. Modify existing hypotheses and form new ones to explain these patterns.
5. Propose and conduct experiments to test the hypotheses.
6. Using the results of these experiments, begin again at step 1.

At every step in this process the choice of what to do next is guided by an internal evaluation of "interestingness". This evaluation is heuristically derived. Heuristics are also used to guide theory formation and experiment planning. Periodically, it may become necessary to synthesize new heuristics.

Lenat set his program to work in a number of domains. Some were already well understood by scientists (e.g. set theory and number theory), while others were relatively new and unexplored (e.g. three-dimensional VLSI circuit design and naval fleet design as applied to the Traveller Trillion Credit Squadron war game). In every case AM/EURISKO was able to make interesting discoveries, although in the case of set and number theory these discoveries were in general

not new. EURISKO designed fleets that were never defeated in the simulated TCS war game. Its work in VLSI design produced plans for many new devices, including one which simultaneously computes NAND and OR in a very simple manner. Its mathematical discoveries included DeMorgan's law, Goldback's conjecture, the fundamental theory of arithmetic, and the concept of prime numbers.

Theories or conjectures in AM/EURISKO are never proved in the strict mathematical sense of the word. Instead, they are validated empirically. As confirming evidence is accumulated, a hypothesis becomes more and more interesting and thus subject to further investigation and refinement. If, instead, disconfirming evidence is discovered, the hypothesis loses interest and is set aside. This reliance on empirical data makes it necessary that the program be able to devise and conduct experiments. Lenat limited his domains to those which could be modeled or simulated internally. To apply EURISKO-like techniques to other areas of science it would be necessary to establish communication links to the outside world, whereby experiments could be proposed and results provided to the program.



## CONCLUSIONS AND RECOMMENDATIONS

Artificial intelligence shows great promise as a way of augmenting traditional computational approaches and as an important tool in problem areas which have so far been intractable to computer solution. Computer scientists must work with scientists in other fields to identify problem areas that can benefit from AI research. The debate as to whether computers can be made to think and act intelligently will continue. This debate is important, for it forces researchers to consider fundamental questions. At the same time, practitioners of AI must not lose sight of the immediate objective, which is to develop processes and techniques that work.

The DENDRAL project shows that expert systems technology can be successfully applied to research problems. Several conditions must be met for a problem to be a good candidate for expert system solution. First, there must be no simple algorithmic solution available. Second, there must be an expert who is willing to contribute his time and expertise to the project. Finally, it should be emphasized that the development of an expert system can be a slow and incremental process. It is appropriate in situations where the pay-off is commensurate with the effort involved.

Future research in expert systems will concentrate on new and better ways of capturing expertise. Improvements in the ability to reason from first principles will allow these systems to solve problems that have not been anticipated during the design phase. This will help alleviate some of the current concern over reliability and verification.

Further advancements in learning theory will permit the development and use of expert systems even in fields which are not well understood. Lenat felt that the AM/EURISKO project would contribute important techniques to this end.

The use of expert systems in conjunction with intelligent tutoring systems is a promising area of research. One of the arguments offered by critics of AI techniques is that excessive reliance on computer-generated solutions may lead to a decrease in human expertise. This possibility could be lessened by using expert systems as both tutors and problem solvers.

Lenat's work in computerized theory formation also offers interesting possibilities. While it is unlikely that computers, unaided, will make important scientific discoveries in the near future, the computer as an intelligent assistant is not an unrealistic goal. Existing technology has already proven its virtues as a fast, reliable calculator. Coupled with intelligent decision making and problem solving techniques, this power can be invaluable.

The ability to analyze large quantities of experimental data intelligently, noting patterns, and suggesting theories to account for them is characteristic of scientific research. A properly trained computer should be able to take advantage of its superior speed and reliability to perform much of the preliminary analysis in this process. Intelligent computers could also be taught to monitor and control experiments within certain limitations. Further developments in some AI fields not specifically covered in this study will also enhance the usefulness of the computer as an intelligent assistant. Chief among these is the area of natural language processing. The ability to communicate easily and directly with a machine will make it much more accessible to the computer-naive user.

The development of intelligent assistants to research scientists must be considered to be a long term and evolutionary process. The knowledge base and the heuristics needed for such an undertaking can only be acquired through experience. Limiting initial work to a narrow domain will allow careful testing of techniques so that eventual expansion of the system will proceed from a firm foundation.

## REFERENCES

- Barr, Avron and Feigenbaum, Edward A., eds., "Applications-oriented AI Research: Science", in The Handbook of Artificial Intelligence, Vol. II, William Kaufmann, Inc., Los Altos, California (1982).
- Brown, J. S., Burton, R., and deKleer, J., "Pedagogical, Natural Language and Knowledge Engineering Techniques in SOPHIE I, II, and III", in Intelligent Tutoring Systems, Sleeman, D. and Brown, J. S., eds. Academic Press, Cambridge, Mass., (1982).
- Buchanan, Bruce G. and Shortliffe, Edward H., Rule Based Expert Systems, Addison-Wesley Publishing Company, Reading, Mass. (1984).
- Bundy, A. Silver, B., and Plummer, B., "An Analytical Comparison of Some Rule-Learning Programs", Artificial Intelligence, Vol. 27, No. 2, 137-181 (1985).
- Charniak, E., and McDermott, D., Introduction to Artificial Intelligence, Addison-Wesley Publishing Company, Reading, Mass., (1985).
- Clancey, W. J., "Dialogue Management for Rule-Based Tutorials", Proceedings of Sixth IJCAI, 155-151 (1979).
- Clancey, W. J., "Tutoring Rules for Guiding a Case Method Dialogue", Intelligent Tutoring Systems, Sleeman, D. and Brown, J. S., eds., Academic Press, Cambridge, Mass. (1982).
- Cohen, Paul, R. and Feigenbaum, Edward A., "Learning and Inductive Inference", in The Handbook of Artificial Intelligence, Vol. III, William Kaufmann, Inc., Los Altos, California (1982).
- Hayes-Roth, Frederick, "The Knowledge-Based Expert System: A Tutorial", Computer, Vol. 17, No. 9, 11-18 (1984).
- Hong, S. J., "Guest Editor's Introduction", Computer, Vol. 19, No. 7, 12-15 (1986).
- Langley, P. W., "Rediscovering Physics with BACON.3", Proceedings of Sixth IJCAI, 505-507 (1979).

- Lenat, D. B., "Theory formation by heuristic search; the nature of heuristics II. Background and examples." Artificial Intelligence, Vol. 21, Nos. 1,2, 31-59 (1983a).
- Lenat, D. B., "EURISKO: A Program That Learns New Heuristics and Domain Concepts. The Nature of Heuristics III." Artificial intelligence, Vol. 21, Nos. 1,2, 61-98 (1983b).
- Matthews, M. and Biswas, G., "Raising User Proficiency Through Active Assistance: An Intelligent Editor", Proceedings Second Conference on Artificial Intelligence Applications, 358-362 (1985).
- McDermott, J. "R1: The formative years", AI Magazine, Vol. 2, No. 2, 21-29 (1981).
- Mitchell, T. M., "Version spaces: A candidate elimination approach to rule learning.", Proceedings of Fifth IJCAI, 305-310 (1977).
- Stevens, A., Collins, A., and Goldin, S. E., "Misconceptions in student's understanding", Intelligent Tutoring Systems, Sleeman, D. and Brown, J. S., eds., Academic Press, Cambridge, Mass (1982).